

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) A method for processing a data packet in an interpartition virtual network in a logical partitioned data processing system, wherein the logical partitioned data processing system comprises a plurality of logical partitions that each simultaneously run an operating system therein, wherein each one of the logical partitions comprises a virtual network adapter that is used to send data packets to other virtual network adapters of other logical partitions within the logical partitioned data processing system to thereby form the interpartition virtual network, the method comprising:

responsive to receiving the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, identifying a state of a first flag and a state of a second flag in the data packet; and

selectively verifying a checksum, by the first partition in the logical partitioned data processing system, for the data packet as indicated by the state of the first flag and the state of the second flag, wherein the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the data packet.

2. (Previously Presented) The method of claim 1, wherein the first flag is a no checksum flag that is used by the selectively verifying step to determine whether or not there is a checksum value included in the data packet that is received and the second flag is a checksum good flag that is used by the selectively verifying step to determine whether or not the data packet has previously been verified as being good based on a checksum included in the data packet that is received.

3. (Previously Presented) The method of claim 2, wherein the selectively verifying step includes: verifying the checksum for the data packet only if the first flag and the second flag are unset.

4. (Previously Presented) The method of claim 2, wherein the selectively verifying step includes: skipping verification of the checksum if the first flag is set.

5. (Previously Presented) The method of claim 2, wherein the selectively verifying step includes: skipping verification of the checksum for the data packet if the first flag is unset and the second flag is set.

6. (Previously Presented) The method of claim 2, wherein the second flag is conditionally unset by the logical partitioned data processing system if the packet was received through a first virtual adapter associated with the first partition.
7. (Previously Presented) The method of claim 2, wherein the first flag is conditionally set by the logical partitioned data processing system if the data packet, received from the second partition, originated from within the logical partitioned data processing system.
8. (Previously Presented) The method of claim 2, wherein the second flag is conditionally unset by the logical partitioned data processing system if the data packet, received from the second partition, was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked.
9. (Previously Presented) The method of claim 8, wherein the first flag is conditionally unset by the logical partitioned data processing system and the second flag is conditionally unset by the logical partitioned data processing system if the data packet was received by a physical network adapter that (i) is associated with the second partition, and (ii) does not support checksum offload.
10. (Previously Presented) The method of claim 8, wherein the first flag is conditionally unset by the logical partitioned data processing system and the second flag is conditionally set by the logical partitioned data processing system if a physical adapter, supporting a checksum offload, verified the checksum as being good.
11. (Previously Presented) The method of claim 1, wherein the data packet is received by a first virtual adapter in the first partition connected to the interpartition virtual network, wherein the first virtual adapter is a software device driver that has no associated physical hardware.
12. (Previously Presented) The method of claim 1 further comprising:
 - generating a new data packet for a target destination;
 - conditionally generating the checksum for the new data packet only if the new data packet is to be sent outside of the interpartition virtual network by a physical network adapter; and
 - sending the new data packet to the target destination.

13. (Previously Presented) The method of claim 1, wherein the first flag and the second flag are added to a header of the data packet that is received by firmware of the logical data processing system during routing of the data packet, to a given partition of the logical partitioned data processing system, by the firmware.

14. (Currently Amended) The method of claim 1, wherein the identifying step comprises identifying a state of a first flag and a state of a second flag in a header in the data packet ~~the first flag and the second flag are located in a header in the data packet that is read by the selectively identifying step to identify the state of the first flag and the state of the second flag.~~

15. (Previously Presented) A logical partitioned data processing system for processing a data packet in an interpartition virtual network in a logical partitioned data processing system, wherein the logical partitioned data processing system comprises a plurality of logical partitions that are each operable for simultaneously running an operating system therein, wherein each one of the logical partitions comprises a data processor for running its respective operating system and a virtual network adapter that is used to send data packets to other virtual network adapters of other logical partitions within the logical partitioned data processing system to thereby form the interpartition virtual network, the data processing system comprising:

identifying means, responsive to receiving the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, for identifying a state of a first flag and a state of a second flag in the data packet; and

selectively verifying means, in a first partition in the logical partitioned data processing system, for selectively verifying a checksum for the data packet as indicated by the state of the first flag and the state of the second flag, wherein the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the data packet.

16. (Currently Amended) The data processing system of claim 15, wherein the selectively verifying means comprises means for selectively verifying a checksum for the data packet as indicated by the state of the first flag and the state of the second flag, wherein the first flag is a no checksum flag located in the data packet that indicates to the selectively verifying means whether or not there is a checksum value included in the data packet that is received, and the second flag is a checksum good flag located in the data packet that indicates to the selectively verifying means whether or not the data packet has previously been verified as being good based on a checksum included in the data packet that is received ~~first flag used by the selectively verifying means is a no checksum flag located in the data packet that indicates~~

~~whether or not there is a checksum value included in the data packet that is received and the second flag used by the selectively verifying means is a checksum good flag located in the data packet that indicates whether or not the data packet has previously been verified as being good based on a checksum included in the data packet that is received.~~

17. (Previously Presented) The data processing system of claim 16, wherein the selectively verifying means includes:

verifying means for verifying the checksum for the data packet only if the first flag and the second flag are unset.

18. (Previously Presented) The data processing system of claim 16, wherein the selectively verifying means includes:

skipping means for skipping verification of the checksum if the first flag is set.

19. (Previously Presented) The data processing system of claim 16, wherein the selectively verifying means includes:

skipping means for skipping verification of the checksum for the data packet if the first flag is unset and the second flag is set.

20. (Previously Presented) The data processing system of claim 16, further comprising means for conditionally unsetting the second flag if the packet was received through a first virtual adapter associated with the first partition.

21. (Previously Presented) The data processing system of claim 16, further comprising means for conditionally setting the first flag if the data packet, received from the second partition, originated from within the logical partitioned data processing system.

22. (Previously Presented) The data processing system of claim 16, further comprising means for conditionally unsetting the second flag if the data packet, received from the second partition, was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked.

23. (Previously Presented) The data processing system of claim 22, further comprising means for conditionally unsetting the first flag and the second flag if the data packet was received by a physical network adapter associated with the second partition.
24. (Previously Presented) The data processing system of claim 22, further comprising means for conditionally unsetting the first flag and conditionally setting the second flag if a physical network adapter, supporting a checksum offload, verified the checksum as being good.
25. (Previously Presented) The data processing system of claim 15, further comprising virtual adapter means for receiving the data packet in the first partition connected to the interpartition virtual network, wherein the virtual adapter means is a software device driver that has no associated physical hardware.
26. (Previously Presented) The data processing system of claim 15 further comprising:
first generating means for generating a new data packet for a target destination;
second generating means for conditionally generating the checksum for the new data packet only if the new data packet is to be sent outside of the interpartition virtual network by a physical network adapter; and
sending means for sending the new data packet to the target destination.
27. (Previously Presented) The data processing system of claim 15, further comprising means for adding, during internal routing of the data packet to a given partition of the logical partitioned data processing system, the first flag and the second flag to a header of the data packet that is received.
28. (Original) The data processing system of claim 15, wherein the first flag and the second flag are located in a header in the data packet.
29. (Previously Presented) A computer program product recorded on a computer readable, recordable-type medium and operable for processing a data packet in an interpartition virtual network-by a logical partitioned data processing system, wherein the logical partitioned data processing system comprises a plurality of logical partitions that are each operable for simultaneously running an operating system therein, wherein each one of the logical partitions comprises a virtual network adapter that is used to send data packets to other virtual network adapters of other logical partitions within the logical

partitioned data processing system to thereby form the interpartition virtual network, the computer program product comprising:

first instructions, responsive to receiving the data packet at a first partition in the interpartition virtual network from a second partition in the interpartition virtual network in the logical partitioned data processing system, for identifying a state of a first flag and a state of a second flag in the data packet; and

second instructions for selectively verifying a checksum, by the first partition in the logical partitioned data processing system, for the data packet as indicated by the state of the first flag and the state of the second flag, wherein the first flag and the second flag are both checksum-based flags that indicate checksum characteristics associated with the data packet.

30. (Currently Amended) The computer program product of claim 29, wherein the first instructions comprise sub-instructions for (i) determining, using the first flag, the first flag is a no-checksum flag that is used by the first instructions to determine whether or not there is a checksum value included in the data packet that is received and ~~the second flag is a checksum good flag that is used by the first instructions to determine~~ (ii) determining, using the second flag, whether or not the data packet has previously been verified as being good based on a checksum included in the data packet that is received.

31. (Previously Presented) The computer program product of claim 30, wherein the second instructions includes:

sub-instructions for verifying the checksum for the data packet only if the first flag and the second flag are unset.

32. (Previously Presented) The computer program product of claim 30, wherein the second instructions includes:

sub-instructions for skipping verification of the checksum if the first flag is set.

33. (Previously Presented) The computer program product of claim 30, wherein the second instructions includes:

sub-instructions for skipping verification of the checksum for the data packet if the first flag is unset and the second flag is set.

34. (Previously Presented) The computer program product of claim 30, further comprising instructions to conditionally unset the second flag if the packet was received through a first virtual network adapter associated with the first partition.

35. (Previously Presented) The computer program product of claim 30, further comprising instructions to conditionally set the first flag if the data packet, received from the second partition, originated from within the logical partitioned data processing system.

36. (Previously Presented) The computer program product of claim 30, further comprising instructions to conditionally unset the second flag if the data packet, received from the second partition, was received from outside the interpartition virtual network in the logical partitioned data processing system without the checksum being checked.

37. (Previously Presented) The computer program product of claim 36, further comprising instructions to conditionally unset the first flag and conditionally unset the second flag if the data packet was received by a physical adapter that (i) is associated with the second partition, and (ii) does not support checksum offload.

38. (Previously Presented) The computer program product of claim 36, further comprising instructions to conditionally unset the first flag and conditionally set the second flag if a physical network adapter, supporting a checksum offload, verified the checksum as being good.

39. (Cancelled)